REMARKS

In this Response, claims 1, 12 and 16-26 have been amended. Claims 1-26 are currently pending, of which claims 1, 12 and 16 are independent. No new matter has been added.

I. Objection to the Drawings

The Examiner objected to the drawings as failing to comply with 37 CFR 1.84(p)(5) because the drawings do not include reference numerals 85, 87, 89, 91 and 95 as mentioned on page 8 of the Specification (office action, paragraph 7). Accordingly, in the amendments to the drawings, Applicants have amended Figure 4 to include reference numerals 85, 87, 89, 91, 93 and 95. No new matter has been added and no new issues have been raised. In view of the amendments to the drawings, Applicants respectfully request reconsideration and withdrawal of the objection to the drawings.

II. Objection to the Claims

The Examiner objected to claims 16-26 because they allegedly do not have proper explicit antecedent basis (office action, paragraph 8). In the claim amendments, Applicants have amended claim 16 so that it recites "said computer-readable medium," rather than "said medium." Accordingly, Applicants have also amended claims 17-26 so that they recite "the computer-readable medium," rather than "the medium." In view of the amendments to claims 16-26, Applicants respectfully request reconsideration and withdrawal of the objection to the claims.

III. Summary of Rejections

Claims 12-26 stand rejected under 35 U.S.C. §101.

Claims 1-4, 6-8, 12-14, 16-19 and 21-23 stand rejected under 35 U.S.C. §102(b) as being anticipated by United States Patent Publication Number 2002/0010908 to Cheng et al (hereafter "Cheng").

Claims 5, 15 and 20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of United States Patent Publication Number US 2004/0085357 to Childress et al (hereafter "Childress").

Claims 9, 10, 24 and 25 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of United States Patent Publication Number US 2003/0225774 to Davidov et al (hereafter "Davidov").

Claim 11 and 26 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cheng and in view of United States Patent Number US 6,066,181 to DeMaster (hereafter "DeMaster").

These rejections will be discussed separately below.

IV. Claim Rejections under 35 USC §101

The Examiner rejected claims 12-26 under 35 U.S.C. §101 as allegedly being directed to non-statutory subject matter (office action, paragraph 10). Applicants respectfully traverse the 35 U.S.C. §101 rejections of claim 12-26 as set forth below.

Regarding claims 12-15, the Examiner alleged that "Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized" (office action, paragraph 10). Amended independent claim 12 recites an electronic device including **a display device** and **a processor**. Applicants believe amended claim 12 to recite statutory subject matter.

Regarding claims 16-26, the Examiner alleged that "the limitation of "said [computer-readable] medium holding instructions" can be reasonably interpreted as the computer-readable medium carrying or transmitting electrical signals, since the instructions are not recorded on the computer-readable medium" (office action, paragraph 10). Amended independent claim 16 recites a computer-readable medium **storing computer-executable instructions**. Applicants believe amended claim 16 to recite statutory subject matter. *See* In re Beauregard, 53 F.3d 1583, 1583-84 (Fed. Cir. 1995).

As such, Applicants respectfully request reconsideration and withdrawal of the outstanding rejection of claims 12-26 under 35 U.S.C. §101.

V. Claim Rejections under 35 USC §102(b)

The Examiner rejected claims 1-4, 6-8, 12-14, 16-19 and 21-23 under 35 U.S.C. §102(b) as being anticipated by Cheng (office action, paragraph 12). Applicants respectfully traverse the 35 U.S.C. §102(b) rejections of claims 1-4, 6-8, 12-14, 16-19 and 21-23 as set forth below.

A. Claim 1

Amended independent claim 1 recites:

> "In an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, a method comprising the steps of:
> providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and
> creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters."

Applicants respectfully submit that the Cheng reference fails to disclose at least the following features of amended independent claim 1: "a custom storage class," and "source code that implements functionality of said graphical model."

i) Summary of the Cheng reference with respect to Claim 1

The Cheng reference relates to a code generation engine that allows a developer to develop new command line interface (CLI) commands quickly and efficiently. Cheng discusses that, in response to a user entering commands at the CLI of an operating system one line after another, the operating system executes the commands in the order of entry (Cheng, paragraph 18). The CLI of Cheng is not a graphical model or a graphical modeling and execution

environment, as recited in claim 1. Cheng discusses representing all the commands available to the user in a command tree (Cheng, paragraph 18). The command tree is a hierarchical representation including command nodes, each command node having an associated handler function (Cheng, paragraph 18). Each handler function has associated handler function code which, when executed, causes the operating system to carry out the command entered by the user (Cheng, paragraph 18). In response to the user entering a command at the CLI, the command tree is traversed starting at the root node to reach an appropriate node (Cheng, paragraph 18). The handler function code associated with the handler function of this node is executed to cause the operating system to carry out the command entered by the user (Cheng, paragraph 18). The command tree of Cheng, like the CLI, is not a graphical model or a graphical modeling and execution environment, as required by claim 1.

Cheng does not disclose several features of claim 1. For example, Cheng does not disclose a custom storage class, as required by claim 1. As described in Applicants' Specification at pages 1-2, a custom data storage class prescribes a manner in which data is represented in software source code produced from a graphical model. Changes to a unique set of instructions defining a custom storage class collectively apply to the set of data of that class. Since Cheng does not disclose a "custom storage class," Cheng cannot disclose "a custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in claim 1.

Cheng does not disclose still other features of claim 1. For example, Cheng does not disclose an automatic code generator that "creates source code that implements functionality of a graphical model," as required by claim 1. Cheng discusses that execution of the handler function code causes the operating system to carry out the command typed in the command-line interface (Cheng, paragraph 18). However, carrying out a command typed in a command-line interface is not equivalent to creating source code that implements functionality of a graphical model. Cheng further does not disclose that execution of the handler function code has anything to do with graphical models, e.g. "implements functionality of a graphical model," as further required by claim 1.

ii) The Examiner's position regarding Claim 1

The Examiner alleges at paragraph 12 of the office action:

> "As per Claim 1, Cheng et al. disclose:
> - providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment (see Figures 4, 6, and 7; Paragraph [0026], "FIG. 7 shows an exemplary GUI 400 for command node editor 120."; Paragraph [0028], "The entering of parameters is also accomplished via GUI 400 by adding the desired parameters to parameter field 410."; Paragraph [0043], "Handler code generation engine 135 automatically generates this software code using the information entered by the developer and the parameter and handler function definitions generated by command structure generation engine 145.");"

The Examiner further alleges in the remarks section at paragraph 17 of the office action:

> "Note that the handler function definitions and parameter definitions are interpreted as "custom storage class," where handler code generation engine automatically generates software code using the information from the handler function definitions and parameter definitions."

iii) Cheng fails to disclose a "custom storage class" as recited in Claim 1

Applicants respectfully submit that Cheng fails to disclose "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in claim 1, because Cheng does not address storage classes. The Examiner points to the handler function definitions and parameter definitions as disclosing the custom storage class recited in claim 1 (Office Action, paragraph 12). Applicants respectfully disagree with the Examiner's characterization of Cheng. Applicants contend that the *handler function definitions and parameter definitions*, discussed in Cheng, are not synonymous with the *custom storage class* recited in claim 1, as set forth below.

As discussed in Applicants' Specification at pages 1-2, each item of data in a graphical model is defined to have a **data storage class**. Data is represented in software source code produced from the graphical model in a manner that is prescribed by its data storage class (Specification, page 1). The software source code references data in a number of different ways including defining data, declaring data, initializing data, reading a value of data, assigning the value of data, and the choice of storage class controls how each of these references are generated (Specification, pages 1-2).

As discussed in Applicants' Specification at page 2, code generators may provide **predefined sets of storage classes**, and they may also permit the user to define new, **custom storage classes** with user-defined characteristics. Changes to the unique set of instructions defining a custom storage class collectively apply to the set of data of that class (Specification, page 2). Common software engineering practices that may be enabled with custom storage classes include, but are not limited to, embedding a data item in a bit field, embedding a data item in a structure, embedding a data item in a union, using platform-specific declarations in the data declaration, defining the scope and storage of the data, declaring data using arbitrary C types, and accessing data through function calls (Specification, page 2).

As discussed above, Cheng discusses that execution of the handler function code associated with the handler function of a command node causes the operating system to carry out the command entered by the user (Cheng, paragraph 18). The handler function code is generated by a handler code generation engine 135 which uses information entered by the developer and parameter and handler function definitions (Cheng, paragraph 43). The parameter definitions and handler function definitions provide information on *how the commands typed in the command-line interface can be carried out* by the operating system. Cheng does not disclose that the parameter definitions or the handler function definitions specify the manner in which handler function code is generated corresponding to *data referenced by said graphical model* in said graphical modeling and execution environment. In fact, Cheng does not mention referencing of data by a graphical model. In contrast, claim 1 requires a custom storage class, with the "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and

execution environment." For example, such source code may include instructions on embedding a data item in a bit field, embedding a data item in a structure, embedding a data item in a union, etc, as described above (Specification, page 2).

iv) Cheng fails to disclose "source code that implements functionality of said graphical model" as recited in Claim 1

Applicants respectfully submit that Cheng fails to disclose "source code that implements functionality of said graphical model," as recited in claim 1. It appears from the Examiner's remarks at paragraph 17 of the office action that the Examiner is pointing to the software code generated by the handler code generation engine in Cheng as disclosing the source code recited in claim 1. As recited in claim 1, execution of the source code executes the graphical model. Cheng does not disclose that execution of the code generated by the handler code generation engine executes a graphical model, as required by claim 1.

The command tree and the CLI of Cheng are not graphical models or graphical modeling and execution environments, as required by claim 1. Cheng teaches that executing the software code associated with a handler function *causes the operating system to carry out the particular command typed by the user in the command-line interface* (Cheng, paragraph 22). Thus, when a user enters a command at the interface, the operating system traverses the command tree branch and reaches an appropriate node (Cheng, paragraph 22). At the node, the operating system retrieves the appropriate handler function and executes the software code associated with the handler function (Cheng, paragraph 22). This results in the operating system *carrying out the command entered by the user* (Cheng, paragraph 22). In contrast, claim 1 requires that execution of the source code *implements the functionality of the graphical model*. Execution of a command entered by a user at a command-line interface, as discussed in Cheng, is not synonymous with implementing the functionality of a graphical model in a graphical modeling and execution environment as recited in claim 1. As such, Cheng fails to disclose "source code that implements functionality of said graphical model," as recited in claim 1.

In view of the above arguments, Applicants respectfully request reconsideration and allowance of claim 1.

B. Claims 2-4 and 6-8

Claims 2-4 and 6-8 depend from independent claim 1 and, as such, incorporate all of the elements of claim 1. Accordingly, claims 2-4 and 6-8 are allowable for at least the reasons set forth above with respect to claim 1. Applicants therefore respectfully request reconsideration and allowance of claims 2-4 and 6-8.

C. Claim 12

Amended independent claim 12 recites:

> "An electronic device having a modeling and execution environment with at least one graphical model, said electronic device comprising:
> a display device for:
> displaying a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model;
> displaying a view of salient aspects of the source code generated by said automatic code generator utilizing the user-selected parameters; and
> a processor for creating a custom storage class in said graphical modeling and execution environment, said custom storage class created utilizing parameters selected by a user from said plurality of selectable parameters."

Applicants respectfully submit that Cheng fails to disclose at least the following feature of claim 12: "a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model." As discussed above in connection with claim 1, Cheng fails to disclose "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." Therefore, Cheng does not support a valid 35 U.S.C. §102(b) rejection of claim 12. Applicants respectfully request reconsideration and allowance of claim 12.

D. Claims 13 and 14

Claims 13 and 14 depend from independent claim 12 and, as such, incorporate all of the elements of claim 12. Accordingly, claims 13 and 14 are allowable for at least the reasons set forth above with respect to claim 12. Applicants therefore respectfully request reconsideration and allowance of claims 13 and 14.

E. Claim 16

Amended independent claim 16 recites:

> "A computer-readable medium for use in an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, said computer-readable medium storing computer-executable instructions for:
> providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and
> creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters."

Applicants respectfully submit that Cheng fails to disclose at least the following feature of claim 16: "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." As discussed above in connection with claim 1, Cheng fails to disclose "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." Therefore, Cheng does not support a valid 35 U.S.C. §102(b) rejection of claim 16. Applicants respectfully request reconsideration and allowance of claim 16.

F. Claims 17-19 and 21-23

Claims 17-19 and 21-23 depend from independent claim 16 and, as such, incorporate all of the elements of claim 16. Accordingly, 17-19 and 21-23 are allowable for at least the reasons set forth above with respect to claim 16. Applicants therefore respectfully request reconsideration and allowance of claims 17-19 and 21-23.

VI. Claim Rejections under 35 USC §103(a)

A. Claims 5, 15 and 20

The Examiner rejected claims 5, 15 and 20 under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of Childress (office action, paragraph 14). Applicants respectfully traverse the 35 U.S.C. §103(a) rejections of claims 5, 15 and 20 as set forth below.

The Cheng reference has been summarized above.

The Childress reference discusses creating, viewing and/or modifying business rules used by an automated insurance claim processing system (Childress, paragraphs [0009-0010]). A rule editor may provide a user with a graphical display of at least a portion of a business rule implemented in software (Childress, paragraphs [0009-0010]). Business rules of knowledge-based system encode the formulas used in evaluating insurance claims in an insurance claim processing software (Childress, paragraph [0007]).

i) Claims 5 and 20

A combination of Cheng and Childress does not disclose or suggest the features of claims 5 and 20. As discussed previously in connection with claim 1, Cheng fails to disclose or suggest the features of claims 1 and 16 from which claims 5 and 20, respectively, depend. For example, Cheng does not disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended

independent claim 1. The teachings of Childress do not supplement Cheng in such a way as to cure the shortcomings of Cheng with respect to the features of independent claims 1 and 16.

For example, regarding claim 1 from which claim 5 depends. Childress fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 1. As such, a combination of Cheng and Childress fails to disclose or suggest the features of claim 5 which depends on claim 1.

Regarding claim 16 from which claim 20 depends, Childress fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 16. As such, a combination of Cheng and Childress fails to disclose or suggest the features of claim 20 which depends on claim 16.

For at least the reasons presented above, Cheng and Childress, alone or in any reasonable combination, fail to disclose or suggest the features of dependent claims 5 and 20. Therefore, the combination of Cheng and Childress does not support a valid 35 U.S.C. §103(a) rejection of claims 5 and 20.

ii) Claim 15

A combination of Cheng and Childress does not disclose or suggest the features of claim 15. As discussed previously in connection with claim 1, Cheng fails to disclose or suggest the features of claim 12 from which claim 15 depends. The teachings of Childress do not supplement Cheng in such a way as to cure the shortcomings of Cheng with respect to the features of independent claim 12.

Regarding claim 12 from which claim 15 depends, Childress fails to disclose or suggest "a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model," as recited in amended independent claim 12. As such, a combination of Cheng and Childress fails to disclose or suggest the features of claim 15 which depends on claim 12.

For at least the reasons presented above, Cheng and Childress, alone or in any reasonable combination, fail to disclose or suggest the features of dependent claim 15. Therefore, the combination of Cheng and Childress does not support a valid 35 U.S.C. §103(a) rejection of claim 15.

B. Claims 9, 10, 24 and 25

The Examiner rejected claims 9, 10, 24, and 25 under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of Davidov (office action, paragraph 15). Applicants respectfully traverse the 35 U.S.C. §103(a) rejections of claims 9, 10, 24 and 25 as set forth below.

The Cheng reference has been summarized above.

A combination of Cheng and Davidov does not disclose or suggest the features of claims 9, 10, 24 and 25. As discussed previously in connection with claim 1, Cheng fails to disclose or suggest the features of claims 1 and 16 from which claims 9, 10, 24 and 25 depend. For example, Cheng does not disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 1.

The Davidov reference discusses an infrastructure for creating applications for mobile information devices, using a tag-based markup language (Davidov, paragraph [0013]). The

teachings of Davidov do not supplement Cheng in such a way as to cure the shortcomings of Cheng with respect to the features of independent claims 1 and 16.

For example, regarding claim 1 from which claims 9 and 10 depend, Davidov fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 1. As such, a combination of Cheng and Davidov fails to disclose or suggest the features of claims 9 and 10 which depend on claim 1.

Regarding claim 16 from which claims 24 and 25 depend, Davidov fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 16. As such, a combination of Cheng and Davidov fails to disclose or suggest the features of claims 24 and 25 which depend on claim 16.

For at least the reasons presented above, Cheng and Davidov, alone or in any reasonable combination, fail to disclose or suggest the features of dependent claims 5 and 20. Therefore, the combination of Cheng and Davidov does not support a valid 35 U.S.C. §103(a) rejection of claims 5 and 20.

C. Claims 11 and 26

The Examiner rejected claims 11 and 26 under 35 U.S.C. 103(a) as being unpatentable over Cheng in view of DeMaster (office action, paragraph 16). Applicants respectfully traverse the 35 U.S.C. §103(a) rejections of claims 11 and 26 as set forth below.

The Cheng reference has been summarized above.

A combination of Cheng and DeMaster does not disclose or suggest the features of claims 11 and 26. As discussed previously in connection with claim 1, Cheng fails to disclose or

suggest the features of claims 1 and 16 from which claims 11 and 26, respectively, depend. For example Cheng does not disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 1.

The DeMaster reference discusses a Java native interface code generator to facilitate mixed language programming (DeMaster, column 2, lines 5-17). The Java native interface code generator makes native code programmed in a native language, such as C, C++ or Assembly, accessible to Java application programs (DeMaster, column 2, lines 5-17). The teachings of DeMaster do not supplement Cheng in such a way as to cure the shortcomings of Cheng with respect to the features of independent claims 1 and 16.

For example, regarding claim 1 from which claim 11 depends, DeMaster fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 1. As such, a combination of Cheng and DeMaster fails to disclose or suggest the features of claim 11 which depends on claim 1.

Regarding claim 16 from which claim 26 depends, DeMaster fails to disclose or suggest "custom storage class specifying the manner in which an automatic code generator creates source code that implements functionality of said graphical model, including source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment," as recited in amended independent claim 16. As such, a combination of Cheng and DeMaster fails to disclose or suggest the features of claim 26 which depends on claim 16.

For at least the reasons presented above, Cheng and DeMaster, alone or in any reasonable combination, fail to disclose or suggest the features of dependent claims 11 and 26. Therefore,

the combination of Cheng and DeMaster does not support a valid 35 U.S.C. §103(a) rejection of claims 11 and 26.

Application No.: 10/698,820 Art Unit: 2191 Docket No.: MWS-062RCE

CONCLUSION

In view of the foregoing amendments and arguments, Applicants believe that all claims should be passed to issuance. Should the Examiner feel that a teleconference would expedite the prosecution of this application, the Examiner is urged to contact the Applicants' attorney at (617) 227-7400.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-062RCE. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: July 24, 2007

Respectfully submitted,

By _____

Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant

Attachments

Annotated Sheet

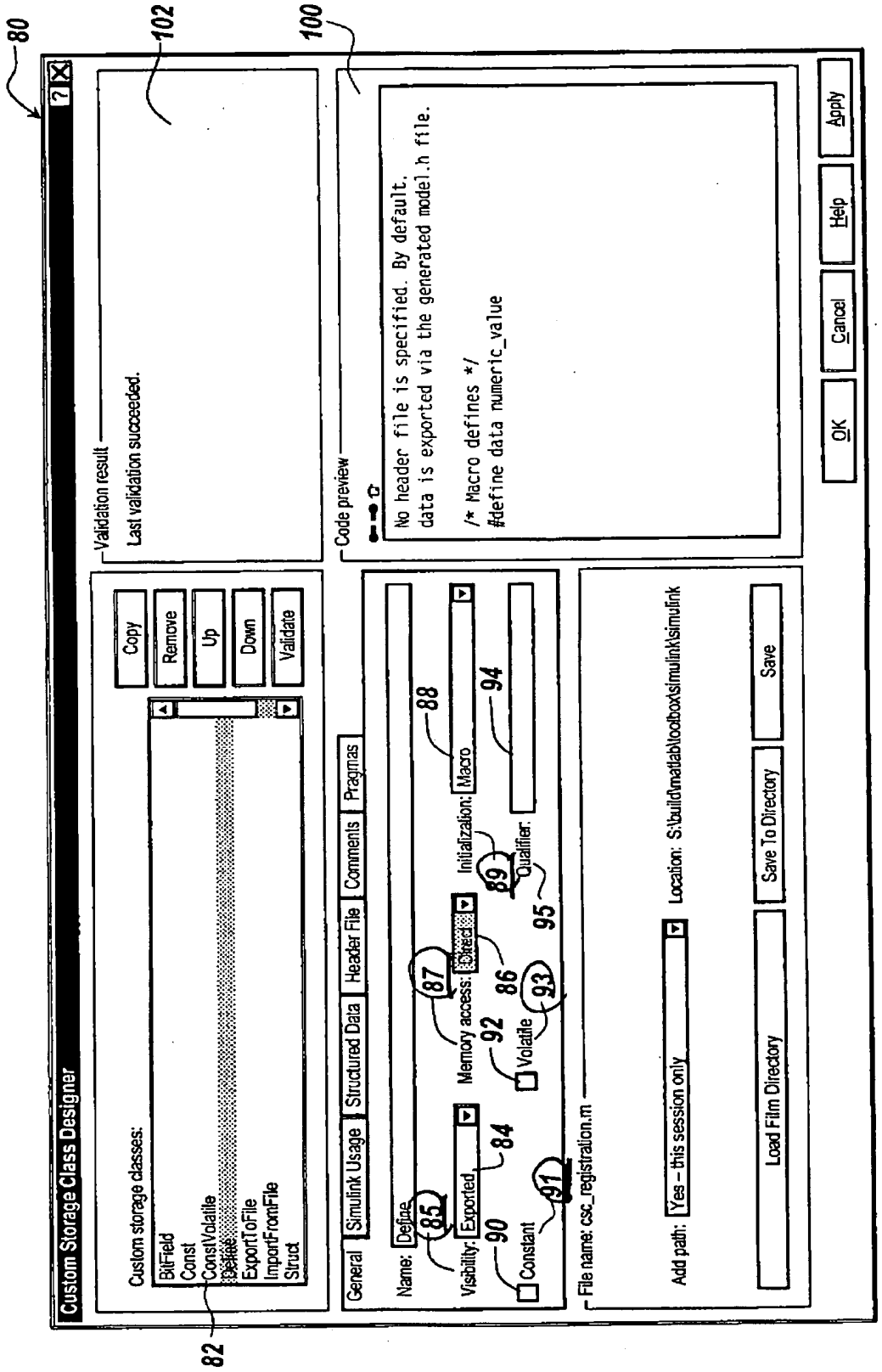

Fig. 4